



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/829,625	04/22/2004	John William Alcorn	ROC920030352US1	7294

7590 05/28/2008  
Robert R. Williams  
IBM Corporation  
Dept. 917  
3605 Highway 52 North  
Rochester, MN 55901-7829

EXAMINER
----------

CHEN, QING

ART UNIT	PAPER NUMBER
----------	--------------

2191

MAIL DATE	DELIVERY MODE
-----------	---------------

05/28/2008

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/829,625	<b>Applicant(s)</b> ALCORN ET AL.	
	<b>Examiner</b> Qing Chen	<b>Art Unit</b> 2191	

**-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --**

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 15 February 2008.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-3,6-8,11,12,15,16,19 and 20 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-3,6-8,11,12,15,16,19 and 20 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)                                | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

### **DETAILED ACTION**

1. This Office action is in response to the RCE filed on February 15, 2008.
2. **Claims 1-3, 6-8, 11, 12, 15, 16, 19, and 20** are pending.
3. **Claims 1, 3, 6, 11, and 16** have been amended.
4. **Claims 4, 5, 9, 10, 13, 14, 17, and 18** have been cancelled.
5. The objection to the specification is withdrawn in view of Applicant's amendments to the specification.
6. The objection to Claim 3 is withdrawn in view of Applicant's amendments to the claim.

### ***Response to Amendment***

### ***Claim Rejections - 35 USC § 103***

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. **Claims 1, 2, 6, 7, 11, 15, 16, and 19** are rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,889,227 (hereinafter "**Hamilton**") in view of US 5,899,990 (hereinafter "**Maritzen**").

As per **Claim 1**, Hamilton discloses:

- receiving a specification of a method in a container-managed persistence bean and a procedure in a backend data store (*see Column 4: 6-10, "The application server receives the database protocol commands or queries from the client computer system and a database bridge converts the database protocol commands to general computer programming language commands of applications running on the application server." and 33-46, "... when the client computer system attempts to access a database field, the request from the client is executed against the created map to determine the corresponding EJB command (e.g. method) for accessing the database field. The EJB command is invoked at the application server and performs the desired operation on the database field desired by the client computer system."*; *Column 6: 22-27, "After the SQL protocol commands are mapped to EJB objects 130, the objects are executed and the EJB objects 130 develop and send queries to the database 110 server 20. A database interface 140 processes the SQL query and accesses the database 150."*);
- generating code in a helper class associated with the container-managed persistence bean, wherein the helper class determines a connector based on a connection factory type (*see Column 6: 14-25, "The database bridge 120 may be defined as a "bridge" class to implement its database protocol command conversion functions." and "The SQL bridge 120 contains a database bridge map 128 that has methods for mapping commands between the client application 104 and EJB objects. After the SQL protocol commands are mapped to EJB objects 130, the objects are executed and the EJB objects 130 develop and send queries to the database 110 server 20."*; *Column 8: 49-53, "After operations defined by EJB logic has been completed at the application server 18, the database is accessed from the application server 18 with a*

Art Unit: 2191

*database protocol language corresponding to the database access request from the client computer system (step 632).”);*

- accessing the procedure via a backend-specific protocol and the connector, wherein the code in the helper class performs the accessing (*see Column 4: 51-56, “The client computer systems 14 communicate in a database access protocol, such as SQL, to the application server 18 ...”; Column 6: 1-9, “OLE-DB is a Microsoft COM API for database access. The Microsoft OLE-DB interface uses database drivers to talk to target databases.” and 14-25, “After the SQL protocol commands are mapped to EJB objects 130, the objects are executed and the EJB objects 130 develop and send queries to the database 110 server 20.”);*

- receiving a specification of input and output records for the procedure (*see Column 7: 28-31, “The SQL commands, as represented in the table, specify a table, T, row, R, column, C, or universal character, such as “\*”, that is used for requesting multiple elements in a single command.”); and*

- mapping the input and output records between the method in the container-managed persistence bean and the procedure, wherein the output records comprise the results, and wherein a state of the container-managed persistence bean persists beyond a lifetime of an application that uses the container-managed persistence bean (*see Figure 4; Column 7: 20-28, “The database bridge map 128 maps EJB methods and properties, used to invoke a designated function in the database, to the SQL commands that perform the functions and to the database elements that are the target of the function. Thus, when the application server 18 receives an SQL command from the client computer system 14, the SQL command is matched to an EJB method that when executed performs the desired operation on the appropriate database*

Art Unit: 2191

*element.” and 31-62, “For example, if a client computer system initiated, from Visual Basic, a “select FirstName” command that command would be converted to an SQL command, such as SQL command 402. The SQL command 402, for example, is a database read command that reads table 1, row 1, column 1. The SQL command 402 is mapped to a “getFirstName” method of a corresponding EJB and the “getFirstName” method is executed at the application server 18.”).*

However, Hamilton does not disclose:

- wherein the code in the helper class calls an evaluator class and passes results of the procedure, wherein the evaluator class evaluates the results.

Maritzen discloses:

- wherein the code in the helper class calls an evaluator class and passes results of the procedure, wherein the evaluator class evaluates the results (*see Column 9: 6-16, “The run() method 310 submits the SQL query to the selected method in step 434.” and “The selected method formats the retrieved columns 440 in accordance with a designation of the received SQL query and stores the retrieved columns in the form of a String.”).*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Maritzen into the teaching of Hamilton to include wherein the code in the helper class calls an evaluator class and passes results of the procedure, wherein the evaluator class evaluates the results. The modification would be obvious because one of ordinary skill in the art would be motivated to format the database information in accordance with the client’s request (*see Maritzen – Column 9: 6-16*).

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and Hamilton further discloses:

- wherein the backend data store comprises a relational database (*see Column 6: 27-31, "The data of an SQL database is relational."*).

As per **Claim 6**, Hamilton discloses:

- means for receiving a specification of a method in a container-managed persistence bean and a procedure in a backend data store, wherein a state of the container-managed persistence bean persists beyond a lifetime of an application that uses the container-managed persistence bean (*see Column 4: 6-10, "The application server receives the database protocol commands or queries from the client computer system and a database bridge converts the database protocol commands to general computer programming language commands of applications running on the application server." and 33-46, "... when the client computer system attempts to access a database field, the request from the client is executed against the created map to determine the corresponding EJB command (e.g. method) for accessing the database field. The EJB command is invoked at the application server and performs the desired operation on the database field desired by the client computer system."; Column 6: 22-27, "After the SQL protocol commands are mapped to EJB objects 130, the objects are executed and the EJB objects 130 develop and send queries to the database 110 server 20. A database interface 140 processes the SQL query and accesses the database 150."*);
- means for generating code in a helper class associated with the container-managed persistence bean, wherein the helper class determines a connector based on a connection factory type (*see Column 6: 14-25, "The database bridge 120 may be defined as a "bridge" class to*

Art Unit: 2191

*implement its database protocol command conversion functions.” and “The SQL bridge 120 contains a database bridge map 128 that has methods for mapping commands between the client application 104 and EJB objects. After the SQL protocol commands are mapped to EJB objects 130, the objects are executed and the EJB objects 130 develop and send queries to the database 110 server 20.”; Column 8: 49-53, “After operations defined by EJB logic has been completed at the application server 18, the database is accessed from the application server 18 with a database protocol language corresponding to the database access request from the client computer system (step 632).”);*

- means for accessing the procedure via a backend-specific protocol and the connector, wherein the code in the helper class performs the means for accessing (*see Column 4: 51-56, “The client computer systems 14 communicate in a database access protocol, such as SQL, to the application server 18 ...”; Column 6: 1-9, “OLE-DB is a Microsoft COM API for database access. The Microsoft OLE-DB interface uses database drivers to talk to target databases.” and 14-25, “After the SQL protocol commands are mapped to EJB objects 130, the objects are executed and the EJB objects 130 develop and send queries to the database 110 server 20.”);*

- means for receiving a specification of input and output records for the procedure (*see Column 7: 28-31, “The SQL commands, as represented in the table, specify a table, T, row, R, column, C, or universal character, such as “\*”, that is used for requesting multiple elements in a single command.”); and*

- means for mapping the input and output records between the method in the container-managed persistence bean and the procedure, wherein the output records comprise the results (*see Figure 4; Column 7: 20-28, “The database bridge map 128 maps EJB methods and*



*properties, used to invoke a designated function in the database, to the SQL commands that perform the functions and to the database elements that are the target of the function. Thus, when the application server 18 receives an SQL command from the client computer system 14, the SQL command is matched to an EJB method that when executed performs the desired operation on the appropriate database element.” and 31-62, “For example, if a client computer system initiated, from Visual Basic, a "select FirstName" command that command would be converted to an SQL command, such as SQL command 402. The SQL command 402, for example, is a database read command that reads table 1, row 1, column 1. The SQL command 402 is mapped to a "getFirstName" method of a corresponding EJB and the "getFirstName" method is executed at the application server 18.”).*

However, Hamilton does not disclose:

- wherein the code in the helper class calls an evaluator class and passes results of the procedure, wherein the evaluator class evaluates the results.

Maritzen discloses:

- wherein the code in the helper class calls an evaluator class and passes results of the procedure, wherein the evaluator class evaluates the results (*see Column 9: 6-16, “The run( ) method 310 submits the SQL query to the selected method in step 434.” and “The selected method formats the retrieved columns 440 in accordance with a designation of the received SQL query and stores the retrieved columns in the form of a String.”).*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Maritzen into the teaching of Hamilton to include wherein the code in the helper class calls an evaluator class and passes results of the

Art Unit: 2191

procedure, wherein the evaluator class evaluates the results. The modification would be obvious because one of ordinary skill in the art would be motivated to format the database information in accordance with the client's request (*see Maritzen – Column 9: 6-16*).

As per **Claim 7**, the rejection of **Claim 6** is incorporated; and Hamilton further discloses:

- wherein the backend data store comprises a relational database (*see Column 6: 27-31, “The data of an SQL database is relational.”*).

**Claims 11 and 15** are storage medium claims corresponding to the apparatus claims above (Claims 6 and 7) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 6 and 7.

As per **Claim 16**, Hamilton discloses:

- a processor (*see Column 5: 35-40, “... the present invention are carried out through the use of a central processing unit (CPU) in conjunction with application programs or modules.”*); and
- a storage device encoded with instructions, wherein the instructions when executed on the processor (*see Column 5: 35-40, “Computer programs and modules used to implement the various steps of the present invention are generally located in a memory unit ...”*) comprise:
  - receiving a specification of a method in a container-managed persistence bean and a procedure in a backend data store, wherein a state of the container-managed persistence bean persists beyond a lifetime of an application that uses the container-managed persistence bean

*(see Column 4: 6-10, "The application server receives the database protocol commands or queries from the client computer system and a database bridge converts the database protocol commands to general computer programming language commands of applications running on the application server." and 33-46, "... when the client computer system attempts to access a database field, the request from the client is executed against the created map to determine the corresponding EJB command (e.g. method) for accessing the database field. The EJB command is invoked at the application server and performs the desired operation on the database field desired by the client computer system."; Column 6: 22-27, "After the SQL protocol commands are mapped to EJB objects 130, the objects are executed and the EJB objects 130 develop and send queries to the database 110 server 20. A database interface 140 processes the SQL query and accesses the database 150."),*

- generating code in a helper class associated with the container-managed persistence bean, wherein the helper class determines a connector based on a connection factory type *(see Column 6: 14-25, "The database bridge 120 may be defined as a "bridge" class to implement its database protocol command conversion functions." and "The SQL bridge 120 contains a database bridge map 128 that has methods for mapping commands between the client application 104 and EJB objects. After the SQL protocol commands are mapped to EJB objects 130, the objects are executed and the EJB objects 130 develop and send queries to the database 110 server 20."; Column 8: 49-53, "After operations defined by EJB logic has been completed at the application server 18, the database is accessed from the application server 18 with a database protocol language corresponding to the database access request from the client computer system (step 632).")*,

Art Unit: 2191

- accessing the procedure via a backend-specific protocol and the connector, wherein the code in the helper class performs the accessing (*see Column 4: 51-56, "The client computer systems 14 communicate in a database access protocol, such as SQL, to the application server 18 ..."; Column 6: 1-9, "OLE-DB is a Microsoft COM API for database access. The Microsoft OLE-DB interface uses database drivers to talk to target databases." and 14-25, "After the SQL protocol commands are mapped to EJB objects 130, the objects are executed and the EJB objects 130 develop and send queries to the database 110 server 20."*),
- receiving a specification of input and output records for the procedure (*see Column 7: 28-31, "The SQL commands, as represented in the table, specify a table, T, row, R, column, C, or universal character, such as "\*", that is used for requesting multiple elements in a single command."*), and
- mapping the input and output records between the method in the contain-managed persistence bean and the procedure, wherein the output records comprise the results (*see Figure 4; Column 7: 20-28, "The database bridge map 128 maps EJB methods and properties, used to invoke a designated function in the database, to the SQL commands that perform the functions and to the database elements that are the target of the function. Thus, when the application server 18 receives an SQL command from the client computer system 14, the SQL command is matched to an EJB method that when executed performs the desired operation on the appropriate database element." and 31-62, "For example, if a client computer system initiated, from Visual Basic, a "select FirstName" command that command would be converted to an SQL command, such as SQL command 402. The SQL command 402, for example, is a database read command that reads table 1, row 1, column 1. The SQL command 402 is mapped to a*

Art Unit: 2191

*"getFirstName" method of a corresponding EJB and the "getFirstName" method is executed at the application server 18.").*

However, Hamilton does not disclose:

- wherein the code in the helper class calls an evaluator class and passes results of the procedure, wherein the evaluator class evaluates the results.

Maritzen discloses:

- wherein the code in the helper class calls an evaluator class and passes results of the procedure, wherein the evaluator class evaluates the results (*see Column 9: 6-16, "The run() method 310 submits the SQL query to the selected method in step 434." and "The selected method formats the retrieved columns 440 in accordance with a designation of the received SQL query and stores the retrieved columns in the form of a String."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Maritzen into the teaching of Hamilton to include wherein the code in the helper class calls an evaluator class and passes results of the procedure, wherein the evaluator class evaluates the results. The modification would be obvious because one of ordinary skill in the art would be motivated to format the database information in accordance with the client's request (*see Maritzen – Column 9: 6-16*).

As per **Claim 19**, the rejection of **Claim 16** is incorporated; and Hamilton further discloses:

- wherein the backend data store comprises a relational database (*see Column 6: 27-31, "The data of an SQL database is relational."*).

9. **Claims 3, 8, 12, and 20** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Hamilton** in view of **Maritzen** as applied to Claims 1, 6, 11, and 16 above, and further in view of **US 6,269,373 (hereinafter “Apte”)**.

As per **Claim 3**, the rejection of **Claim 1** is incorporated; however, Hamilton and Maritzen do not disclose:

- wherein the backend data store comprises a non-relational database.

Apte discloses:

- wherein the backend data store comprises a non-relational database (*see Column 6: 54-57, “The above mentioned methods could be written to access other backend systems (i.e. CICS, IMS, MQ, SAP, etc.) and should not be restricted to just JDBC or database access.”*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Apte into the teaching of Hamilton to include wherein the backend data store comprises a non-relational database. The modification would be obvious because one of ordinary skill in the art would be motivated to utilize and access non-relational databases (*see Apte – Column 6: 54-57*).

As per **Claim 8**, the rejection of **Claim 6** is incorporated; however, Hamilton and Maritzen do not disclose:

- wherein the backend data store comprises a non-relational database.

Apte discloses:

Art Unit: 2191

- wherein the backend data store comprises a non-relational database (*see Column 6: 54-57, "The above mentioned methods could be written to access other backend systems (i.e. CICS, IMS, MQ, SAP, etc.) and should not be restricted to just JDBC or database access."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Apte into the teaching of Hamilton to include wherein the backend data store comprises a non-relational database. The modification would be obvious because one of ordinary skill in the art would be motivated to utilize and access non-relational databases (*see Apte – Column 6: 54-57*).

**Claim 12** is rejected for the same reason set forth in the rejection of Claim 8.

As per **Claim 20**, the rejection of **Claim 16** is incorporated; however, Hamilton and Maritzen do not disclose:

- wherein the backend data store comprises a non-relational database.

Apte discloses:

- wherein the backend data store comprises a non-relational database (*see Column 6: 54-57, "The above mentioned methods could be written to access other backend systems (i.e. CICS, IMS, MQ, SAP, etc.) and should not be restricted to just JDBC or database access."*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Apte into the teaching of Hamilton to include wherein the backend data store comprises a non-relational database. The modification would be

Art Unit: 2191

obvious because one of ordinary skill in the art would be motivated to utilize and access non-relational databases (*see Apte – Column 6: 54-57*).

### ***Response to Arguments***

10. Applicant's arguments with respect to Claims 1, 6, 11, and 16 have been considered, but are moot in view of the new ground(s) of rejection.

#### ***In the Remarks, Applicant argues:***

a) Thus, Hamilton does not teach or suggest "generating code in a helper class associated with the container-managed persistence bean, wherein the helper class determines a connector based on a connection factory type; ...wherein the code in the helper class calls an evaluator class and passes results of the procedure, wherein the evaluator class evaluates the results," as recited in claim 1 because neither the Hamilton SQL Query Model 124 (which merely "process[es] queries") nor the Hamilton "Database Bridge Map 128" (which includes data that describes a mapping relationship) determines a connector based on a connection factory type, calls an evaluator class, or pass results of the procedure, as recited in claim 1.

#### ***Examiner's response:***

a) Examiner disagrees. Applicant's arguments are not persuasive for at least the following reasons:

First, Hamilton clearly discloses "generating code in a helper class associated with the container-managed persistence bean, wherein the helper class determines a connector based on a



Art Unit: 2191

connection factory type” (see Column 6: 14-25, “The database bridge 120 may be defined as a “bridge” class to implement its database protocol command conversion functions.” and “The SQL bridge 120 contains a database bridge map 128 that has methods for mapping commands between the client application 104 and EJB objects. After the SQL protocol commands are mapped to EJB objects 130, the objects are executed and the EJB objects 130 develop and send queries to the database 110 server 20.”; Column 8: 49-53, “After operations defined by EJB logic has been completed at the application server 18, the database is accessed from the application server 18 with a database protocol language corresponding to the database access request from the client computer system (step 632).”). Note that the database bridge contains the SQL query model and the database bridge map. The database bridge (helper class) maps the SQL commands to the EJB objects (generates code associated with the container-managed persistence bean). The application server then accesses the database with a database protocol language by executing the EJB objects (helper class determines a connector based on a connection factory type). Thus, in effect, the database bridge of the application server is a layer of abstraction between the client application and the database.

Second, the claims recite only “a connector based on a connection factory type” with no further clarification on the claim scope of the term “connector” as intended by the Applicant to cover. Thus, as the claims are interpreted as broadly as their terms reasonably allow (see MPEP § 2111.01 I), the interpretation of a broad limitation of “connector” as a database protocol and the like by one of ordinary skill in the art is considered to be reasonable by its plain meaning.

***In the Remarks, Applicant argues:***

b) The Office Action relies on the Hamilton at column 6, lines 2-25, 32-24, and 43- 51, which merely recite that commands are mapped to EJB objects and that methods are mapped as properties in the database bridge, none of which is related to mapping output records that comprise results, as recited in claim 1 because commands and methods are not output results. Thus, Hamilton does not teach or suggest "mapping the input and output records between the method in the container-managed persistence bean and the procedure, wherein the output records comprise the results," as recited in claim 1.

***Examiner's response:***

b) Examiner disagrees. Applicant's arguments are not persuasive for at least the following reasons:

First, Hamilton clearly discloses "mapping the input and output records between the method in the container-managed persistence bean and the procedure, wherein the output records comprise the results" (*see Figure 4; Column 7: 20-28, "The database bridge map 128 maps EJB methods and properties, used to invoke a designated function in the database, to the SQL commands that perform the functions and to the database elements that are the target of the function. Thus, when the application server 18 receives an SQL command from the client computer system 14, the SQL command is matched to an EJB method that when executed performs the desired operation on the appropriate database element."* and 31-62, "For example, if a client computer system initiated, from Visual Basic, a "select FirstName" command that command would be converted to an SQL command, such as SQL command 402. The SQL command 402, for example, is a database read command that reads table 1, row 1, column 1.

Art Unit: 2191

*The SQL command 402 is mapped to a "getFirstName" method of a corresponding EJB and the "getFirstName" method is executed at the application server 18.*”). Note that a SQL command specifies the output. For example, the SQL command specifies the output data in table 1, row 1, and column 1, which is the first name of an employee. Thus, when an SQL command is mapped to an EJB method (mapping the input records), the output data of the SQL command is also mapped to the output data of the EJB method, since both the SQL command and the EJB method refer to the same output.

Second, the claims recite only “output records” with no further clarification on the claim scope of the limitation “output records” as intended by the Applicant to cover. Thus, as the claims are interpreted as broadly as their terms reasonably allow (see MPEP § 2111.01 I), the interpretation of a broad limitation of “output records” as output data and the like by one of ordinary skill in the art is considered to be reasonable by its plain meaning.

### ***Conclusion***

11. The prior art made of record and not relied upon is considered pertinent to Applicant’s disclosure.

12. Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

Art Unit: 2191

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/QC/

April 1, 2008

/Wei Zhen/

Supervisory Patent Examiner, Art Unit 2191